

### **DETAILED ACTION**

1. This office action is in response to correspondence filed 10/29/09 regarding application 10/812999, in which claims 1, 10-12, and 17 were amended. Claims 1 and 3-17 are pending in the application and have been considered.

### ***Response to Arguments***

2. The arguments on pages 6-9 of the Remarks have been considered, but are moot in view of the new grounds of rejection, which were necessitated by the amendment to claims 1, 10-12, and 17.

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1, 3, and 5-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Law et al. (2006/0025997) in view of De Brabander (2004/0243387).

Consider claim 1, Law discloses a computer implemented method of generating a spoken dialog application (**Fig 2**), comprising:  
generating a no more than two-dimensional graphical representation of a call flow which does not alter finite state machines in real time (**Fig 4, [0046]**)

generating a grammar representation of the call flow using the two-dimensional graphical representation ([0049], [0052], [0061], the grammar generated for one or more specific IVR platforms, [0064]).

Law does not specifically mention a *context free* grammar, generating a finite state machine from the context free grammar representation of the call flow, the finite state machine comprising a plurality of nodes including at least a first leaf node and at least a first root node; and

generating a dialogue application code for a spoken dialogue application from said finite state machine, wherein said generated dialogue application code for said functions are executable during runtime of said spoken dialogue application for walking the finite state machine from the at least one root to the at least one leaf of the finite state machine.

De Brabander discloses generating a finite state machine from a context free grammar representation of a language model, the finite state machine comprising a plurality of nodes including at least a first leaf node and at least a first root node ([0065], [0115], the FSMs are modeled with RTNs, which are coded in Backus-Naur Format); and

generating a language model application code for a language model application from said finite state machine ([0061], compiling to create software), wherein said generating language model application code for said functions are executable during runtime of said language model application for walking the finite state machine from the at least one root to the at least one leaf of the finite state machine ([0617], links traverse

the FSM).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law by generating a finite stat machine from the context free grammar representation of the call flow, the finite state machine comprising a plurality of nodes including at least a first leaf node and at least a first root node; and generating a dialogue application code for a spoken dialogue application from said finite state machine, wherein said generated dialogue application code for said functions are executable during runtime of said spoken dialogue application for walking the finite state machine from the at least one root to the at least one leaf of the finite state machine, in order to facilitate making the language development process less prone to mistakes by providing immediate feedback, as suggested by De Brabander **(Col 1 lines 8-15)**.

Claim 10 is directed to a computer-readable medium for implementing the method of claim 1, and so is rejected for similar reasons.

Consider claim 11, Law discloses a processor in communication with a module, wherein the module is configured to generate a grammar representation of a call flow generated using a no more than two-dimensional graphical representation of the call flow, wherein generating the two dimensional graphical representation does not alter finite state machines in real time **([0049], [0052], [0061]**, the grammar generated for one or more specific IVR platforms, **[0064]**).

Law does not specifically mention generating a finite state machine from a context free grammar representation, wherein the finite state machine comprises a plurality of nodes including at least a first leaf node and at least a first root node, wherein the module is configured to generate application code using said finite state machine, wherein the application dialogue code is generated dependent on how said finite state machine is traversed, for functions to be executed upon state transitions in said generated finite state machine, wherein said generated application code for said functions are executable during runtime of said spoken dialog application wherein the finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine.

De Brabander discloses generating a finite state machine ([0065], [0115]) from a context free grammar representation of a language model ([0071-0072]), wherein the finite state machine comprises a plurality of nodes including at least a first leaf node and at least a first root node ([0065], [0115]); and wherein the module is configured to generate application code using said finite state machine ([0061], compiling to create software), wherein the application code is generated dependent on how said finite state machine is traversed ([0617]), for functions to be executed upon state transitions in said generated finite state machine ([0617]), wherein said generated application code for said functions are executable during runtime of said application ([0061]), wherein the finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine ([0617]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law by generating a finite state machine from a context free grammar representation, wherein the finite state machine comprises a plurality of nodes including at least a first leaf node and at least a first root node, wherein the module is configured to generate application code using said finite state machine, wherein the application dialogue code is generated dependent on how said finite state machine is traversed, for functions to be executed upon state transitions in said generated finite state machine, wherein said generated application code for said functions are executable during runtime of said spoken dialog application wherein the finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine, for reasons similar to those of claim 1.

Consider claim 12, Law discloses a spoken dialog application method comprising:

generating a grammar representation of a call flow generated using a no more than two-dimensional graphical representation of the call flow, wherein generating the two-dimensional graphical representation does not alter finite state machines in real time ([0049], [0052], [0061], the grammar generated for one or more specific IVR platforms, [0064]).

Law does not specifically mention traversing a finite state machine, that is generated from a context free grammar representation of a language model and comprises at least a first root node and at least a first leaf node; generating application

code as said finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine, and invoking said generated application code for functions associated with nodes in said finite state machine, wherein each node of said finite state machine is mapped to a corresponding function.

De Brabander discloses traversing a finite state machine, that is generated from a context free grammar representation of a language model **([0071-0072])** and comprises at least a first root node and at least a first leaf node **([0617])**; generating application code as said finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine **([0617])**, and invoking said generated application code for functions associated with nodes in said finite state machine **([0061])**, wherein each node of said finite state machine is mapped to a corresponding function **([0066]**, a state associated with a parse tree implies that state is mapped to a parsing function).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law by traversing a finite state machine, that is generated from a context free grammar representation of a language model and comprises at least a first root node and at least a first leaf node; generating application code as said finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine, and invoking said generated application code for functions associated with nodes in said finite state machine, wherein each node of said finite state machine is mapped to a corresponding function, for reasons similar to those of claim 1.

Claim 17 is directed to a system for implementing the method of claim 12, and so is rejected for similar reasons.

Consider claim 3, Law discloses the graphical representation is generated using standardized graphical elements (**Fig 4**).

Consider claims 5 and 14, Law does not, but De Brabander discloses the context free grammar representation is in a Backus-Naur Form format (**[0072]**).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law such that the context free grammar representation is in a Backus-Naur Form format, in order to benefit from pre-existing grammar libraries already coded in the well known format.

Consider claims 6 and 15, Law does not, but De Brabander suggests the context free grammar representation is in an augmented Backus-Naur Form format (**[0071-0072]**, RTN is an extension of context free grammar, which suggests an extension, or at least an augmentation of Backus-Naur).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law such that the context free grammar representation is in an augmented Backus-Naur Form format for similar reasons to those of claim 5.

Consider claim 7, Law does not, but De Brabander discloses a function is associated with a node in said finite state machine ([0387], a Function to add a looping 3DTransition between two different states).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law such that a function is associated with a node in said finite state machine for reasons similar to those of claim 1.

Consider claim 8, Law discloses customizing generated application code (Fig 3, step 606).

Consider claims 9 and 16, Law discloses generated application code associated with an output function performs a table lookup prompt information ([0088]).

Consider claim 13, Law discloses said grammar representation is generated from a graphical representation of a call flow ([0049], [0052], [0061], the grammar generated for one or more specific IVR platforms, [0064]).

Law does not specifically mention context free grammar.

De Brabander discloses context free grammar ([0072]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law to include context free grammar for reasons similar to those of claims 1 and 5.

5. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Law et al. (2006/0025997) in view of De Brabander (2004/0243387), in further view of Yuschik (7,139,706).

Consider claim 4, Law and De Brabander do not specifically mention the graphical representation is generated using VISIO.

Yuschik discloses a graphical representation is generated using VISIO (**Col 14 lines 48-51**).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the invention of Law and De Brabander such that a graphical representation is generated using VISIO, since it is a desirable platform for decoupling user interface issues posed by the prompting structure, as suggested by Yuschik (**Col 14 lines 46-48**).

### ***Conclusion***

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jesse Pullias whose telephone number is 571/270-5135. The examiner can normally be reached on M-F 9:00 AM - 4:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Hudspeth can be reached on 571/272-7843. The fax phone number for the organization where this application or proceeding is assigned is 571/270-6135.

8. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Application/Control Number: 10/812,999

Page 12

Art Unit: 2626

/Talivaldis Ivars Smits/

Primary Examiner, Art Unit 2626

1/8/2010